WHAT IS CLAIMED:

1. A modulo addressable data path register file for a processor, comprising:

a first set of registers (RD_X) and

a second set of registers (RI_X);

where the first set of registers stores addresses of the second set of registers and where the second set of registers stores data ; and

where two or more of the first set of registers are ordered sequentially in a circular structure such that the first register falls next in sequence after the last.

2. The register file of claim 1,

where the registers in the circular structure change their contents according to the equation RD_X = RD_(X+k) (modulo M, where k is an integer, each time a processor loop begins a new iteration; and

where the modulus M is equal to the number of registers from the first set used in the circular structure multiplied by |k|, for nonzero k, and by 1 for k=0.

3. The register file of claim 2, where k is one of 0, +/- 1, +/- 2, +/-3 or +/- 4.

4. The register file of claim 3, where the N registers in the first set are numbered from 0 to N-1.

5. The register file of claim 4, where the circular structure is used to store a sequence of N data samples, each being delayed one sample period from the prior sample in the sequence.

6. The register file of claim 5, where the parameter N is programmable at a start-up of processor operation, and is equal to one greater than that the maximum supported delay, as expressed in units of sample periods.

7. The register file of claim 6, where the samples are stored in the RI_X register set.

8. The register file of claim 7, where the RI_X registers are pointed to by the RD_X registers in the circular structure.

9. The register file of claim 8, where the samples are stored in sequential locations in the RI_X register set.

10. A multi processor system, comprising:

a plurality of cells, each with an individual processor;

where each cell has the register file of claim 6, and

where the processor can execute instructions whose operands are the RD_X registers.

11. The system of claim 10, where each cell has a programmable parameter which sets the value of N for that cell.

12. A method of optimizing digital signal processing, comprising;

implementing modulo addressing in a first register bank (RD_X);

enabling the processor to operate on data in a second register bank by operating on a register that points to the data.

13. The method of claim 12, where

the registers in the first register bank change their contents according to either the equation RD_X = RD_(X+k) (modulo M), where k is an integer, each time a processor loop begins a new iteration; and

where the modulus M is equal to a number equal to the registers in the first register bank used to point to data in the second register bank multiplied by |k| for nonzero k, and by 1 for k=0.

14. The method of claim 13, where an unused register in either the first or the second register banks stores the value of M.

15. The method of claim 14, where a dedicated register in the first register bank stores the value of M.

16. A method of implementing digital filtering, comprising:

storing a current data sample and a number of prior data samples in a first register bank;

indexing said current sample and prior data samples by the relative delay to the current sample; and

automatically updating the contents of the first register bank each sample period to write a new data sample over the most delayed sample stored in the register bank.

17. The method of claim 16, where the indexing of the data samples in the first register bank is maintained by a second register bank (RD_X) which stores the addresses of the registers in the first register bank.

18. The method of claim 17, where the second register bank is automatically incremented each sample period according to the equation RD_X = RD_(X+k) (modulo M), where k is an integer, each time a processor loop begins a new iteration; and

where the modulus M is equal to a number equal to the registers in the first register bank used to point to data in the second register bank multiplied by k multiplied by |k|, for nonzero k, and by 1 for k=0.

19. A method of implementing digital filtering, comprising:

storing a first data set comprising a current data sample and a number of prior data samples in a first register bank;

storing one or more additional data sets, each comprising a current data sample and a number of prior data samples in an additional register bank;

indexing each said data set by the relative delay of a sample to the current sample; and

automatically updating the contents of each of the first register bank and the additional register banks each sample period to write a new data sample over the most delayed sample stored in each register bank.

20. The method of claim 19, where the indexing of the data samples in the first register bank and each of the additional register banks is maintained by a pointer register bank (RD_X) which stores the addresses of the registers in the first and each of the additional register banks.

21. The method of claim 20, where the pointer register bank is automatically incremented each sample period according to the equation RD_X = RD_(X+k) (modulo M), where k is an integer, each time a processor loop begins a new iteration; and
where the modulus M is equal to a number equal to the registers in the pointer register bank used to point to data in the first and each of the additional register banks, multiplied by |k|, for nonzero k, and by 1 for k=0.